ARL-CR-0827 ● Apr 2018

US Army Research Laboratory

Vids: Version 2.0 Alpha Visualization Engine

prepared by Gregory Shearer and Joshua Edwards
ICF
9300 Lee Hwy
Fairfax, Virginia 22031

under contract W911QX-17-C-0018

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-CR-0827 ● APR 2018

US Army Research Laboratory

Vids: Version 2.0 Alpha Visualization Engine

by Gregory Shearer and Joshua Edwards
ICF
9300 Lee Hwy
Fairfax, Virginia 22031

under contract W911QX-17-C-0018

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* April 2018	**2. REPORT TYPE** Contractor Report	**3. DATES COVERED (From - To)** September 2016–March 2018

4. TITLE AND SUBTITLE Vids: Version 2.0 Alpha Visualization Engine	**5a. CONTRACT NUMBER** W911QX-17-C-0018
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Gregory Shearer and Joshua Edwards	**5d. PROJECT NUMBER** W911QX-17-C-0018
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) ICF 9300 Lee Hwy Fairfax, VA 22031	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-CIN-D 2800 Powder Mill Road Adelphi, MD 20783-1138	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) ARL-CR-0827

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Visualization for cyber security awareness can be used by cyber security analysts and decision makers to assess trends and patterns in large volumes of network traffic information, aiding situational understanding and analysis capabilities. A need exists for new forms of digital visualization, enabling higher levels of interactivity and visual fidelity than existing efforts. Vids is a project aimed at producing more dynamic and interactive visualization tools using modern computer game development technology to enable faster, more in-depth, and more immersive visualization of cyber security data and events.

15. SUBJECT TERMS visualization, visualization tools, 3-D visualizations, virtual reality, cyber security, situational awareness, traffic analysis, threat analysis

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Gregory Shearer
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	30	**19b. TELEPHONE NUMBER (Include area code)** (301) 394-4617

Contents

List of Figures

Acknowledgments

The authors would like to acknowledge the contributions of the following people to the project: Lee Trossbach, for project management and feedback; Curtis Arnold, for project support and oversight; and Kaur Kullman, for idea generation and feedback over the course of the project, as well as technical collaboration.

INTENTIONALLY LEFT BLANK.

1. Introduction

Visualization is a key element of battlefield planning and awareness for the Warfighter. Traditional visualization uses maps, organizational charts, and analytics to plan, rehearse, and execute operations as part of intelligence preparation of the battlefield performed before, during, and after military operations (Wade 2005). In cyberspace, the terrain of the battlefield is constantly changing. In place of traditional terrain such as rivers and mountains, networks and other communication infrastructure become key terrain features in the cyber domain. Given the different terrain, new visualizations and visualization methods are needed to facilitate cyber security situational understanding.

Visualization for cyber security awareness can be used by cyber security analysts and decision makers to assess trends and patterns in large volumes of network traffic information, potentially faster than any other form of information media such as textual lists or spreadsheets. Network traffic and organization visualization thus can be a key tool to building an ability to understand a rapidly evolving and complex network environment. In the near future, emerging technologies including virtual reality (VR), augmented reality (AR), and mixed reality will likely become more widespread. These technologies may create a fundamental shift in the way data are currently visualized. In the context of intelligence preparation of the battlefield, VR and AR may allow decision makers to process larger volumes of information far more quickly than is possible using traditional methods. Rather than looking at a paper map, commanders may be able to place themselves in a virtual representation of the battlefield, or in a virtualized representation of communication networks rather than a network diagram. To explore this new visualization environment, new tools are needed to translate network data into 3-D visualizations, enable interactivity with these visualizations, and integrate the visualizations into an analyst's workflow.

Vids is a 3-D visualization tool under development by the US Army Research Laboratory (ARL) to fill the 3-D visualization gap, building off prior publication (Zage and Zage 2010) and patent (Trossbach and Pino 2015) work. Vids projects raw text data into a 3-D environment and allows users to move through and interact with the data to improve informational understanding. The Vids software leverages off-the-shelf modern game development technology using the Unity[1] development platform with the aim of enabling faster development than a ground-up solution,

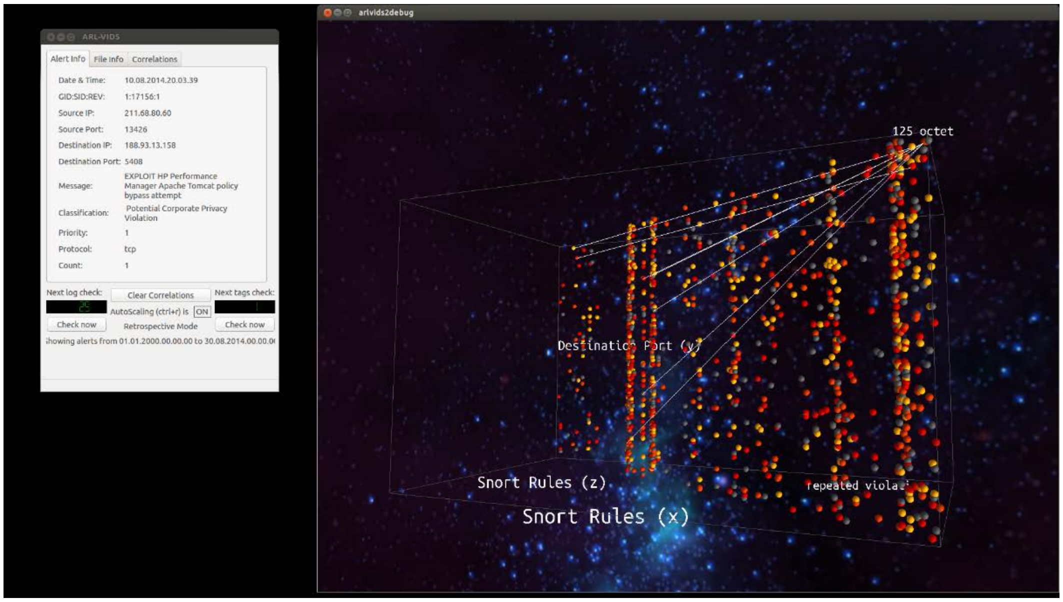
[1] https://unity3d.com/

and allowing for multiplatform compatibility through targeted builds for different operating systems including Windows and Linux.

2. History

The software discussed in this report builds on the ideas of an earlier joint visualization project between ARL and Ball State University (BSU). Originally named the Visual Intrusion Detection System (VIDS), the software focused solely on visualizing Intrusion Detection System (IDS) alerts in maneuverable 3-D space, without consideration for VR or AR. The IDS alerts were placed in a structured 3-D scatterplot for cyber security analysts to quickly view alerts, categorize alerts based on location in the graph, and find correlations between alerts.

To render its 3-D visualizations, the original VIDS (Fig. 1) used the Ogre3D engine for its visualizations and Qt for the user interface. The 2 tools provided the necessities for 3-D graphics rendering and fulfilled a cross-platform requirement for both Windows and Linux support. The project explored different rendering methods, examined processor and memory performance limits of visualization, and investigated user preferences for elements such as backgrounds and illumination.



Date & Time: 10.08.2014.20.03.39
GID:SID:REV: 1:17156:1
Source IP: 211.68.80.60
Source Port: 13426
Destination IP: 188.93.13.158
Destination Port: 5408
Message: EXPLOIT HP Performance Manager Apache Tomcat policy bypass attempt
Classification: Potential Corporate Privacy Violation
Priority: 1
Protocol: tcp
Count: 1

AutoScaling (ctrl+r) is ON
Retrospective Mode
howing alerts from 01.01.2000.00.00.00 to 30.08.2014.00.00.0

Fig. 1 Appearance of the original VIDS

The ARL/BSU joint project creating VIDS ran for approximately 4 years before concluding in 2014. The final product contained all of the requested features but was hampered by unknown crashes, a rigid code structure that prevented further feature additions without significant rewriting, and the loss of knowledge after most of the original collaborative development team either graduated from universities or moved on to new projects.

VIDS was presented to several cyber security analysts at ARL for testing and gathering opinions. The sessions were informal and not meant to be scientifically rigorous, and exact comments were not recorded. Discussions at the time indicated that analysts appreciated the scatterplot layout and its ability to accentuate alert patterns. Negative comments mentioned VIDS' crashing and the lack of features the analysts wanted.

With most of the original VIDS team leaving and feature requests coming from analysts, the decision was made to develop an updated version of the software internally at ARL using the lessons learned from VIDS. The new software name, Vids, no longer an abbreviation, expands beyond solely visualizing IDS alerts and intends to give analysts more freedom in deciding what data they will bring into the 3-D space for analysis and correlations.

Recently, an ARL-pioneered project named Virtual Reality Data Analysis Environment (VRDAE) used VR and a modern game engine to enable multiperson collaboration in a virtual environment. Vids functions as a parallel project to VRDAE, exploring specific problems within the 3-D visualization research space. The intent is not to redevelop or duplicate work done on VRDAE; rather, Vids primarily aims to create a nonVR-focused platform for operationally useful data reading, extraction, and rendering in a 3-D environment. In the future, elements or lessons learned from Vids may be integrated into VRDAE, and in-development knowledge sharing will take place between the respective development teams to leverage lessons from designing the respective environments.

3. Purpose

The purpose of the new iteration of Vids is to provide an interactive 3-D environment for visualizing network and alert data using Unity, a state-of-the-art game development platform and engine. The initial alpha iteration of Vids provides a platform for further development and research on 3-D visualization for network analysis. Later versions are aimed at refining the techniques, visualization compatibility in a VR or AR environment, and active integration into the analyst's workflow. The advantages that may be realized in a 3-D visualization environment include additional dimensionality for displaying information, the ability to change perspective, and the ability to produce an immersive environment. The main drawbacks are issues with depth in visualizations (i.e., occlusion of data) and the additional computing performance required to display a visualization in a 3-D environment compared to a 2-D environment.

The Vids project is aimed at addressing open questions in logical layout of network features into a 3-D visualization. The question of which layouts are best suited to network security needs is an open one. A number of studies (Ferebee 2008; Shiravi 2012) suggest a variety of different layouts with no clearly superior layout for specific goals such as network situational awareness. Vids aims to allow a high degree of flexibility for users by organizing data into any number of available layouts, while allowing users to transition between layout states without loading screens.

Another significant question Vids intends to address is how analysts can best interact with data inside a 3-D visualization environment. Specifically, Vids seeks to investigate what interactions are feasible and, through the mechanism of analyst feedback, what interaction mechanisms are desirable, including functions such as filtering data, sorting data, moving objects, and changing visual styles.

By providing a platform to investigate these questions, Vids is intended as a foundation for several areas of further research. From a basic research perspective, Vids can be used as a platform for evaluating what metrics of visualization utility are useful to the analyst or Warfighter. Vids can also be used to evaluate what cyber symbology and iconography is most effective for conveying meaning to analysts and decision makers. Additionally, as a tool, Vids can be used as it is for visualizing a variety of data or it may be tailored in the future to specific visualization tasks according to operational needs.

4. Tool Components/Body

The current development level for Vids is considered as “alpha”, meaning significant feature development is still underway. The majority of the fundamental basic features are complete, but some will require more refinement and others may be added in the future as needs are further defined and user feedback is received.

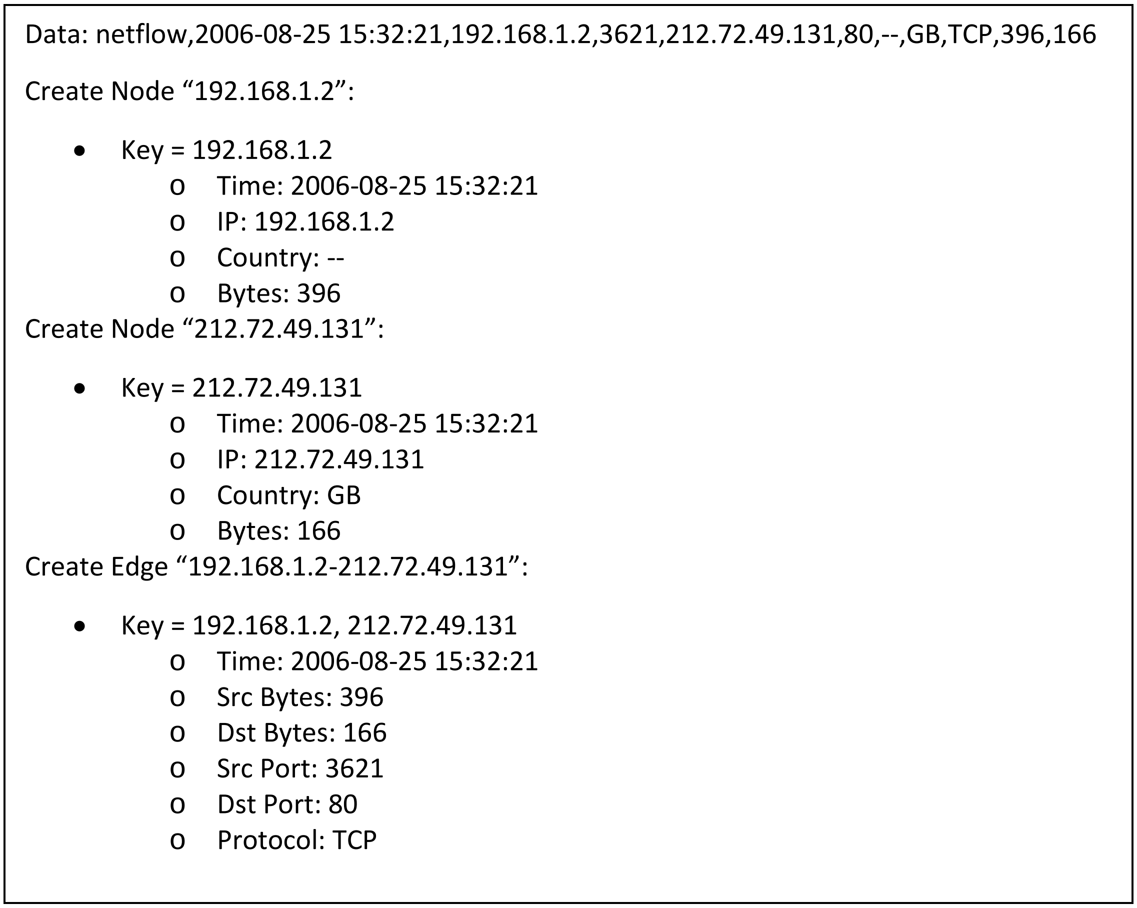
4.1 Data Input

The alpha version of Vids currently reads stored data only from comma separated value (CSV) files. In the future, a JavaScript Object Notation (JSON) data reading module will read data from JSON files in a similar way. A CSV file may be static, may be updated by local processes, or may be the output of a database operation. To visualize data in Vids, the user specifies the location of a CSV file and a data configuration file. Vids then reads the CSV data and processes the raw data into nodes and edges according to the rules defined in the configuration file.

In essence, the data for both nodes and edges are handled in the same way in terms of configuration. The data configuration structure (Fig. 2) defines a unique key field or set of key fields for each type of node and edge to create. In addition, each node or edge type defines a set of attribute fields, specified by CSV column index(es), to collect each time the unique key appears in the CSV file. These attribute fields may include string, numeric, IP address, and/or date/time data.

The data processing process does not necessarily create a new node for every row of input. A single line of data, in this case a netflow record produced using the tool DShell[2], can be used to create several node and edge objects. One or more key fields are defined by the configuration file and these fields specify how node and edge data should be grouped. For example, if multiple flows are observed that involve the IP address 192.168.1.2, the time and number of bytes will be aggregated and recorded in a common data object rather than a unique object per appearance (see Fig. 2). This strategy has the effect of reducing the typical number of nodes that must be rendered and allows for some natural aggregations, such as quickly summing the total number of bytes from or to a given IP.

[2] An extensible network forensic analysis framework. https://github.com/USArmyResearchLab/Dshell



Data: netflow,2006-08-25 15:32:21,192.168.1.2,3621,212.72.49.131,80,--,GB,TCP,396,166

Create Node “192.168.1.2”:

- Key = 192.168.1.2
    - Time: 2006-08-25 15:32:21
    - IP: 192.168.1.2
    - Country: --
    - Bytes: 396

Create Node “212.72.49.131”:

- Key = 212.72.49.131
    - Time: 2006-08-25 15:32:21
    - IP: 212.72.49.131
    - Country: GB
    - Bytes: 166

Create Edge “192.168.1.2-212.72.49.131”:

- Key = 192.168.1.2, 212.72.49.131
    - Time: 2006-08-25 15:32:21
    - Src Bytes: 396
    - Dst Bytes: 166
    - Src Port: 3621
    - Dst Port: 80
    - Protocol: TCP

Fig. 2 Example breakdown of the CSV data parsing methodology

Data processing configuration also specifies the styles to be applied to the data. For example, textures can be assigned to nodes based on IP address, colors can be assigned to nodes and edges either randomly or according to a mapping, and the size of nodes and edges can be assigned according to linear or logarithmic scales.

4.2 Data Views

The Vids alpha version provides a variety of data views, currently 8 different major types, some with additional subtypes. These are presented to the user as a set of selectable layouts that dictate how data are arranged within the virtual 3-D environment. Each data view has parameters that can be adjusted by the end user. Such parameters include algorithmic details, such as the desired radius of a randomly arranged sphere layout or the repulsion versus attraction coefficient of a force-directed graph layout, and feature selection details, such as which data features should be plotted on the x, y, and z axes, or which features should be used to form groups of nodes.

The current set of graph layout options for data views are as follows:

- Spherical and cubic volumes in which nodes are placed either in regular fashion in equally spaced intervals randomly distributed within the volume, or distributed on the outer surface of the volume.
- Nodes ordered along xy-axes based on user-selectable attributes with regular node spacing and node stacking in the z-axis. Node position on the x-axis is determined by the magnitude of the node's x-axis mapped attribute relative to the magnitude of the attribute in other nodes within the graph. Node y-axis position is determined similarly using the y-axis mapped attribute, if one exists. Nodes with the same x-axis and y-axis position can either overlap or be stacked (placed at regular intervals) in the z-axis. The spacing between nodes is controlled by the available space in the graph or by a specified minimum value.
- Nodes placed in a scatterplot along x-, y-, and z-axes based on user-selectable attributes. Nodes are placed in position according to the magnitude of their axis-mapped attributes and are placed according to the selected scaling method for each axis, either linear scaled or logarithmically scaled. Node stacking, in which nodes with overlapping positions can be placed at regular intervals in another axis, is possible in the x- or z-dimension depending on what other axes are already mapped to attributes. If all x-, y-, and z-axes are in use for mapping, stacking will not occur and nodes will overlap by default.
- Force-directed algorithm plotting (Hu 2005). This algorithm is principally used for node-edge networks in which no preexisting structure is present. In this algorithm, nodes are represented as repulsors while edges are represented as attractors (springs) connecting nodes, and the system is allowed to settle to an equilibrium over a series of iterations modeling the spring–repulsor interaction of the nodes and edges.
- Group plotting of nodes based on user-selectable attribute. Nodes are collected into groups based upon the value of a specified attribute. These groups are then displayed in a number of different layout formulations as follows:
    - Histogram chart, placing nodes along the xy-axis of the graph where the x-axis position is an individual group and the y-axis indicates the overall number of nodes belonging to each group.

  - Pie chart, acting similar to the histogram chart in the xy-axis of the graph, but grouping the nodes into slices of a pie chart.
  - Spatial grouping of nodes in xyz-space within the graph. Nodes within a group are placed as tight clusters at different locations in 3-D space. This method is sometimes called “galaxy” visualization.

- Geographical globe plotting of locations based on longitude and latitude coordinates. Nodes are assigned a location based on their indicated physical location on earth. A globe map of earth is used as a prop for this layout. Edges curve outward from the surface to connect nodes.
- Circular plots by degree of connectedness. Nodes are plotted as a ring with edges plotted in the interior of the ring. The nodes are ordered by the number of edges that connect to or from them.
- Hierarchy plot based on direction of edge connections between nodes, similar to a cone-tree plot. Source nodes, nodes that only have edges directed outward, are placed at the top of the plot in a large radius circle and subsequent child nodes connected to these nodes are placed on smaller radius circles under their respective parent node.

All layouts are composed of a combination of node and edge objects. The general appearance of node and edge objects within the Vids platform is shown and labelled in Fig 3. A discussion of nodes, edges, and the various customizable components of each object appears in the next 2 sections (Sections 4.3 and 4.4).

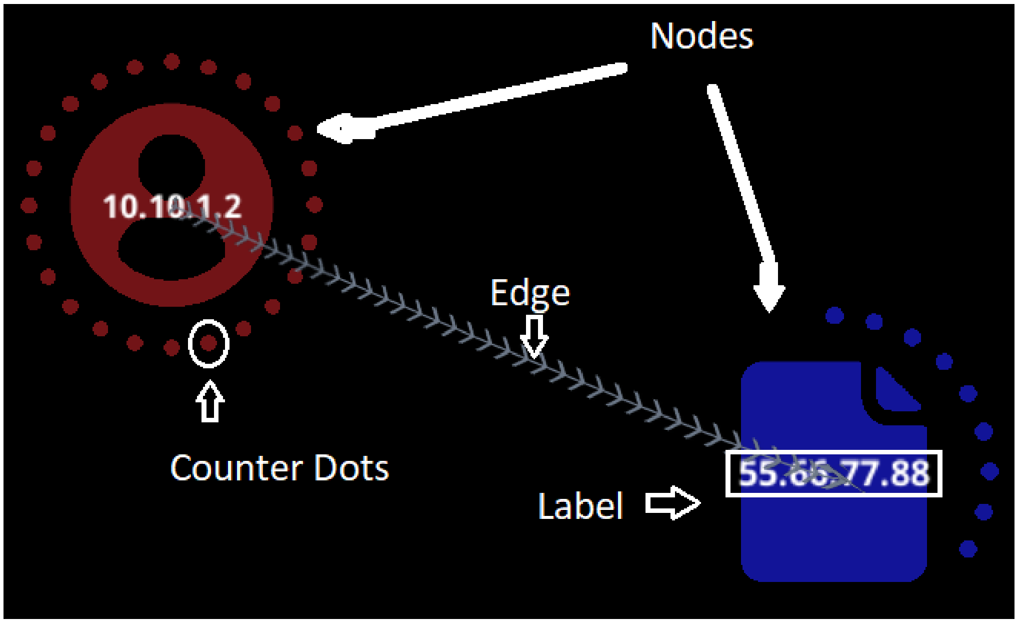


Fig. 3 Definition of node and edge in the context of Vids, and some labelled key features

4.3 Nodes

Nodes in Vids represent a basic entity that can exist in a graph independent of other objects. In essence, nodes in Vids function as they do in any standard node–edge graph. Nodes can be interacted with by the users via selection, detail examination, and movement functions. The rendering style of the individual nodes is configurable in 8 different ways via style parameters. The style parameters may be configured by the user or the developer of the visualization to reflect the data the node holds. For example, node colors can be assigned to represent different groups of data and the node texture may be an icon for the type of node represented. These style parameters include the following:

- Shape of node (i.e., 3-D model used to represent the node)
- Scaled size of node
- Texture mapped to node shape
- Color (in the Unity platform, the "albedo" color) applied over node texture
- Material (including glow effects) with which the texture is rendered
- Node label text
- Rotation rate of node
- Node color pulse rate

Nodes also have a built-in counter feature to provide a visual indication of the number of times a data key appears in the data. In addition to the shape, texture, and color used to represent the node, Vids includes an additional ring of up to 24 dots around the node object, based on a count of how often the entity represented by the node appears in the data. This feature allows all nodes to be consistently sized, which reduces incidence of node overlap or other accidental occlusion of nodes within a visualization.

4.4 Edges

Edges in Vids relate a node to another node. Edges function in a similar manner in Vids as they do in a standard node-edge graph. They link nodes according to specific start and end key parameters. In addition, edges can hold informational attributes unique to each edge. For example, an edge representing a network data flow from host to host would likely hold information about the number of bytes and packets observed in that flow. Users interact with edges in a similar way to nodes: they can be selected, examined, and moved. Edges also have style parameters similar to nodes. The exact set of parameters is subject to change during development, but active and planned features currently include the following:

- Edge thickness
- Edge start and end color, displayed as a gradient if they differ
- Edge texture
- Edge material
- Edge direction indicator density

4.5 Interaction

A critical component of Vids is the ability for users to interact with the displayed data and the environment in which the data is displayed. Interaction allows the user to more closely examine the data, either in aggregate or in detail. The viewpoint can be rotated to view a graph at new angles, possibly revealing data that would have been obscured in a 2-D representation. Interaction also allows users to further scrutinize the data to obtain additional computational information not usually available in a graph visualization, including aggregated data, sums, averages, and maximums/minimums.

In Vids, every node or edge belongs to a graph. These graphs are selectable. Any click on a node or edge object belonging to the graph will select the graph, and a second click on the node or edge object selects the node or edge object. More than one object can be selected at a time by drawing a box around the objects using left click, or by using key-based multiselection. When one or more objects are selected, a display window can be opened that includes both individual information for each node and aggregated information from all selected objects. The aggregated information is collected from each node and edge’s attributes’ data.

Selected objects can be manually moved to different locations. Users can click and drag nodes freely through space, allowing them to organize data as needed. Edge movement is handled by moving the control point for the quadratic Bézier curve traced by the displayed edge. Clicking and dragging a graph can move it, allowing the creation of custom dashboards for multiple data sources. In addition to basic selection features, some additional features exist for selected objects.

Vids alpha includes a generic and always accessible *Main Menu* for creating graphs, listing and selecting graphs, displaying help information, and quitting the application. When one or more graphs are selected, the menu expands to include a *Graph Menu* containing 3 major graph-specific actions: setting the layout of the selected graph, setting the styles (color and texture mapping, sizing, line directionality, etc.) used on nodes and edges within the selected graph, and setting the filter (time interval, value range, value list) used to screen data in the graph. When one or more nodes or edges are selected, the menu expands again to include a *Selection Menu* containing options to tag and correlate objects. The tag option effectively allows users to add a custom comment to the object label and to color code the label as desired. The correlate option is a form of search function that performs a search and selects other nodes and edges that share specific attribute values.

5. Applications/Visuals

5.1 Alerts

VIDS, the previous version of Vids, primarily focused on visualization for the purpose of enhancing intrusion-event detection situational awareness using visual cues. The original concept was to map event properties—such as IP address information, port information, and rule ID number of alerts—to a plot used for awareness and “at a glance” trend analysis of alert patterns. As a follow-on to this original version, the new version of Vids can also be used to perform alert visualization by mapping alert properties to x-, y-, and z-axes of Cartesian coordinates in the 3-D virtual volume.

Figure 4 shows 2 hierarchy graphs of network intrusion detection alerts. To the left, IP addresses are rendered as nodes, while alerts are rendered as edges, with hierarchy position dependent on which malicious IP last communicated with the target IP. To the right, a more strict hierarchy is maintained, consisting of sensor at the top level, followed by tool at the middle layer, followed by individual alerts at the lower level. To an analyst or decision maker, this information may be useful in network mapping, network profiling, and intrusion awareness.

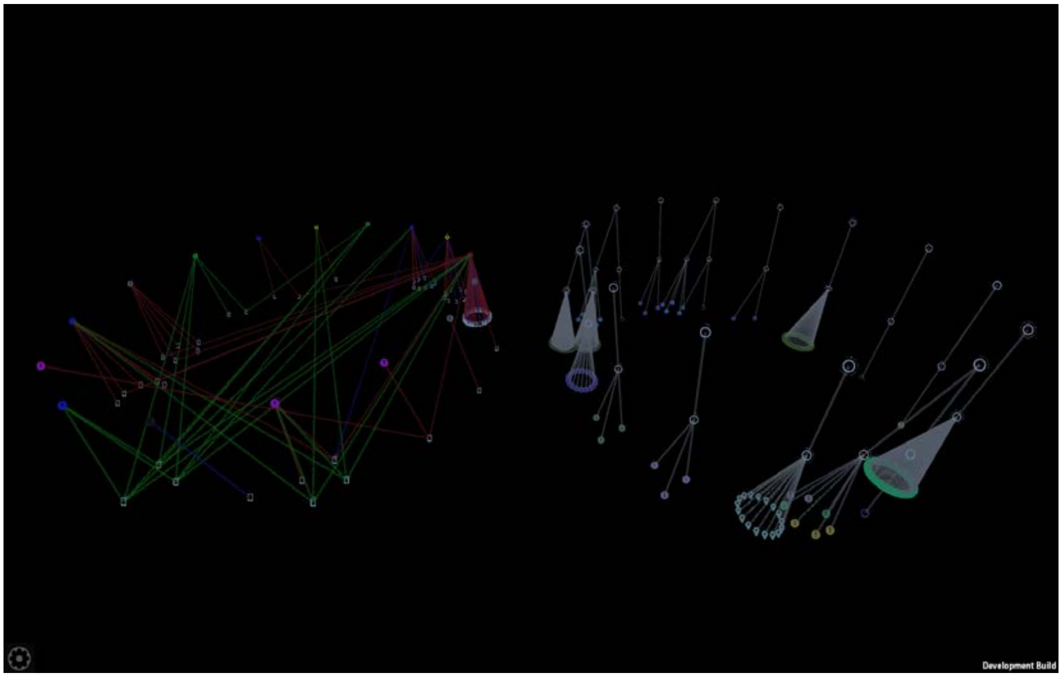


Fig. 4 Hierarchical representation of alerts in 2 different formats

5.2 Flows and Directed Graphs

Vids has the ability to view a network scenario or network traffic as a directed node-edge graph. The rationale for this addition of capability compared to the older VIDS project is to provide a means of visualizing flow data and similar node-edge type graphs in a format that may be more intuitive for analysts and decision makers. Further, a directed acyclic node-edge graph can be used for denoting hierarchical relationships, for example, a representation of a network topology or clear linking between a specific host or IP and all its communication partners.

Figure 5 shows flows from a Skype session. The packet capture of the traffic was obtained at the Wireshark public packet capture sample page.[3] The visualized network is strongly centralized around the initiator of the flow, that is, the Skype participant on whose network the capture was conducted. A large number of outbound connections can be observed from the red node at right (the participant's machine) and a variety of IP addresses located in many different countries.

[3] SkypeIRC.cap. https://wiki.wireshark.org/SampleCaptures

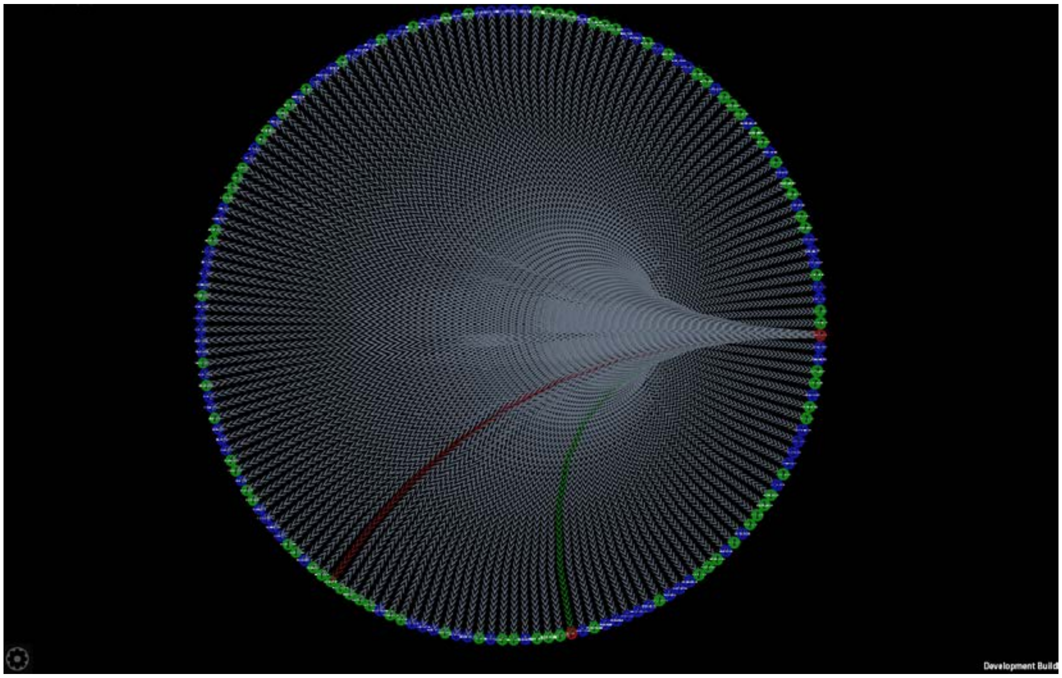


Fig. 5 Circular chart ordered clockwise by degree from 3 o'clock clockwise

5.3 Case Study Example Using Applied Research and Experimentation Partner Data Sets (CyberVAN)

A specific use case of Vids alpha is visualizing the various experiments of the ARL Cyber Security Applied Research and Experimentation Partner (AREP) data sets, produced by Applied Communication Sciences (Vencore Labs, Basking Ridge, New Jersey) in collaboration with ARL (Alberts et al. 2015), also sometimes known as the cyber virtual ad hoc network (CyberVAN) data sets. The CyberVAN data sets include extensive traffic capture from a simulated malware infection event. The host-to-host communication sessions can be extracted based on IP address, enriched with information, and treated as net flows from host to host. Therefore, a node-edge graph can be created in Vids based on the CyberVAN data. Example visualizations originate from Data Sets A Version 1 and B Version 1.

Figure 6 contains a visualization of CyberVAN Data Set A Version 1 (on the left) and CyberVAN Experiment B Version 1 (on the right) with flows arranged using a force-directed layout algorithm. Using this algorithm automatically produces a tree-like structure based on the number and arrangement of node–edge connections via edges. Note that these are 2 separate graphs rather than a single connected graph. Nodes represent IP addresses; edges represent communication flows (transmission control protocol, user datagram protocol sessions, etc.) between the IP addresses. The scenarios depict a simulated network intrusion in 2 variations: A and B. For an analyst or decision maker, these visualizations may provide a faster method of visualizing the activity on the network as a whole.

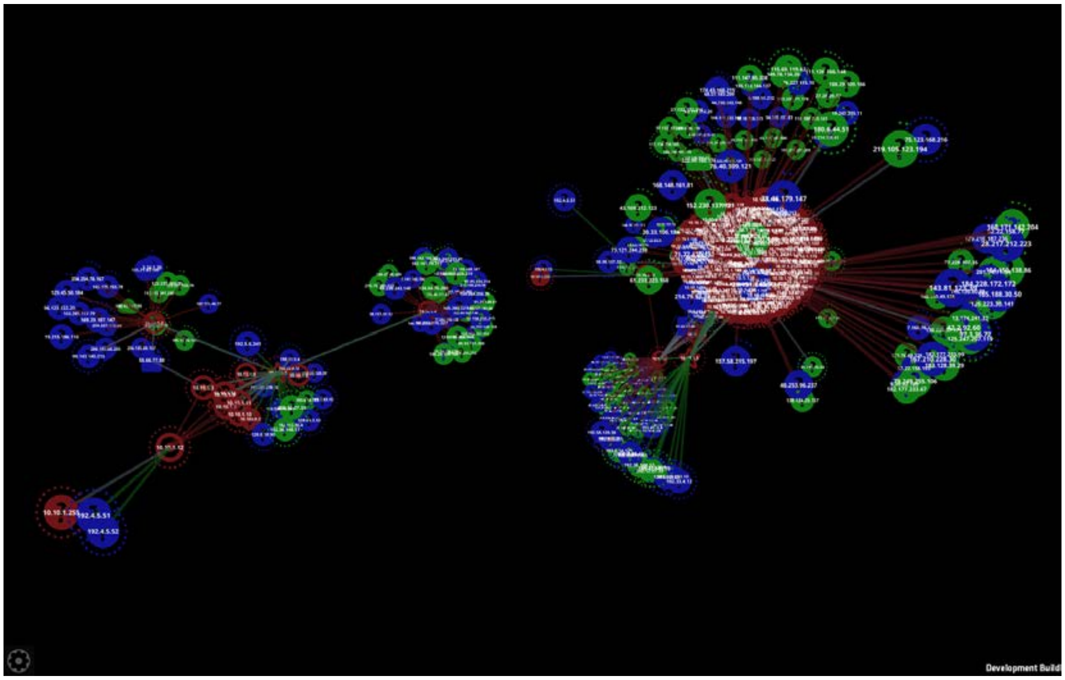


Fig. 6 Simultaneous rendering of 2 CyberVAN data sets: force directed

As in Fig. 6, Fig. 7 shows CyberVAN Experiment A Version 1 data on the left and CyberVAN Experiment B Version 1 data on the right. In this view, edges (representing flows) have been hidden, leaving only IP address nodes. The data sets are arranged in different layouts using the country code parameter (as mapped by a geo-IP database) contained in each data point. A variety of layouts can be applied to the data to demonstrate different types of visualizations.

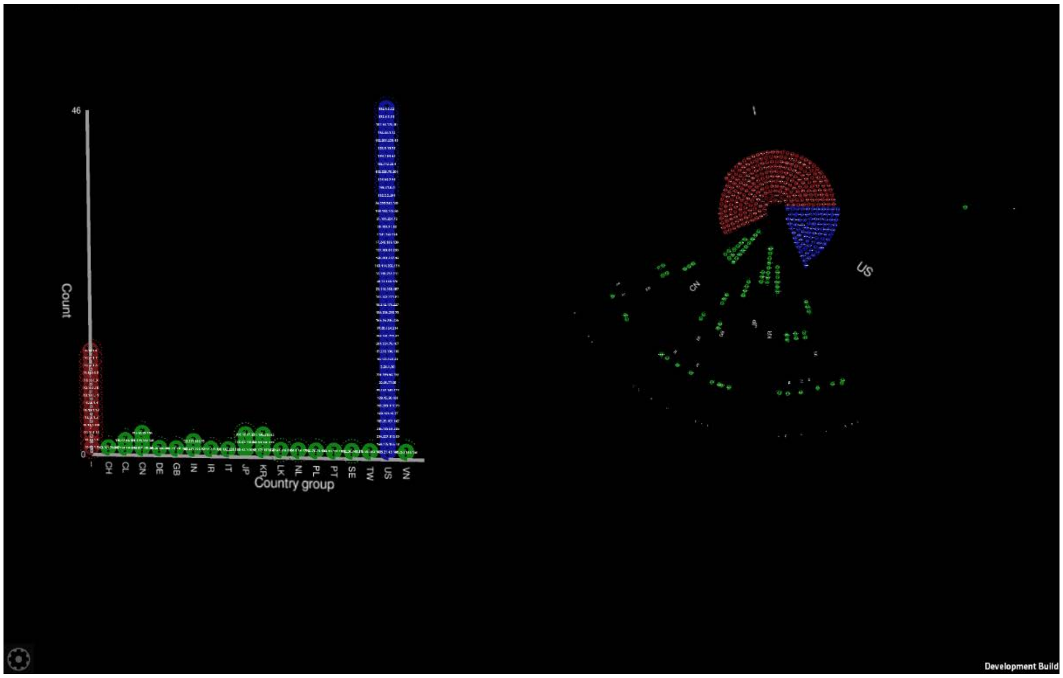


Fig. 7 Simultaneous rendering of 2 CyberVAN data sets: layout variety

Vids allows a complete view of the evolution of a CyberVAN scenario, including a visualization of the situation at distinct points in time. For example, filters can be applied to limit the data to a specific time window, or through filters show experiment progress from start to finish. Data deemed irrelevant can be hidden. For example, if traffic to US-mapped IP addresses is deemed irrelevant, that data can be filtered out and hidden from view. For an analyst or decision maker, the ability to reconfigure the data into traditional bar and pie charts may be useful in understanding key statistics about the data under examination.

As in Figs. 6 and 7, Fig. 8 shows CyberVAN Experiment A Version 1 data on the left and CyberVAN Experiment B Version 1 data on the right. Only IP address node objects are shown in this view. On the left is a scatterplot plotting the number of bytes seen in flows to a given IP address on the y-axis over time on the x-axis. If more than one IP address node object has the same number of bytes and time information, the extra nodes are plotted in the z-axis. On the right is an ordered plot of how often each IP address appeared in the data set, where nodes with the same count are stacked, creating a frequency distribution plot. Figure 9 shows CyberVAN Experiment A in a circular-by-degree layout, similar to Fig. 5 but using different data.

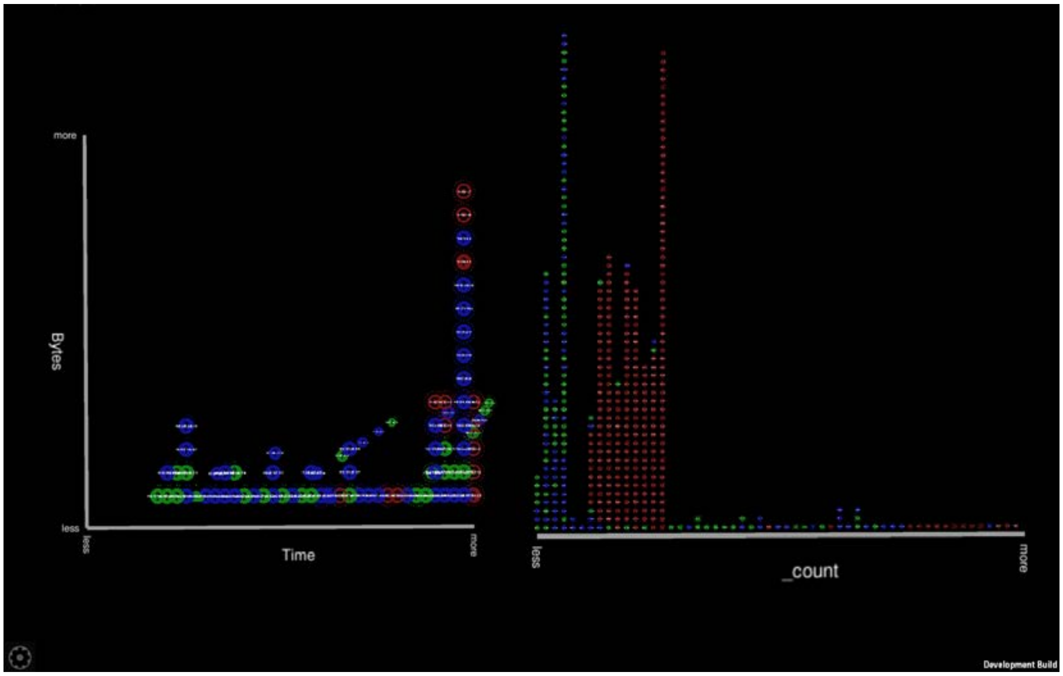


Fig. 8 Simultaneous rendering of 2 CyberVAN data sets: scatterplot and ordered

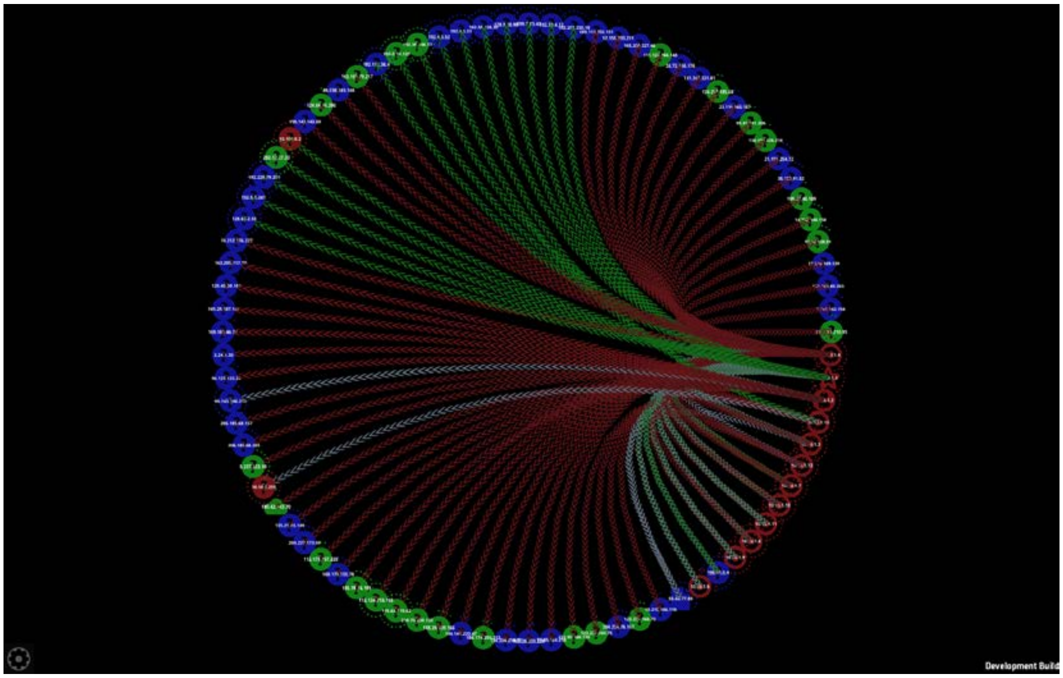


Fig. 9 Circular-by-degree visualization of CyberVAN Data Set A Version 1

5.4 Other Use Cases

Vids can be used for generalized data visualization beyond node-edge graphs. As a test case, web log data was captured, processed into a vectorized representation, and clustered using t-distributed stochastic neighbor embedding (Fig. 10). The results of this clustering were plotted in Vids and enriched with metadata to help find and define apparent clusters in the observed data. By comparing the color scheme with the apparent clustering, the relative similarity of each color group can be compared.

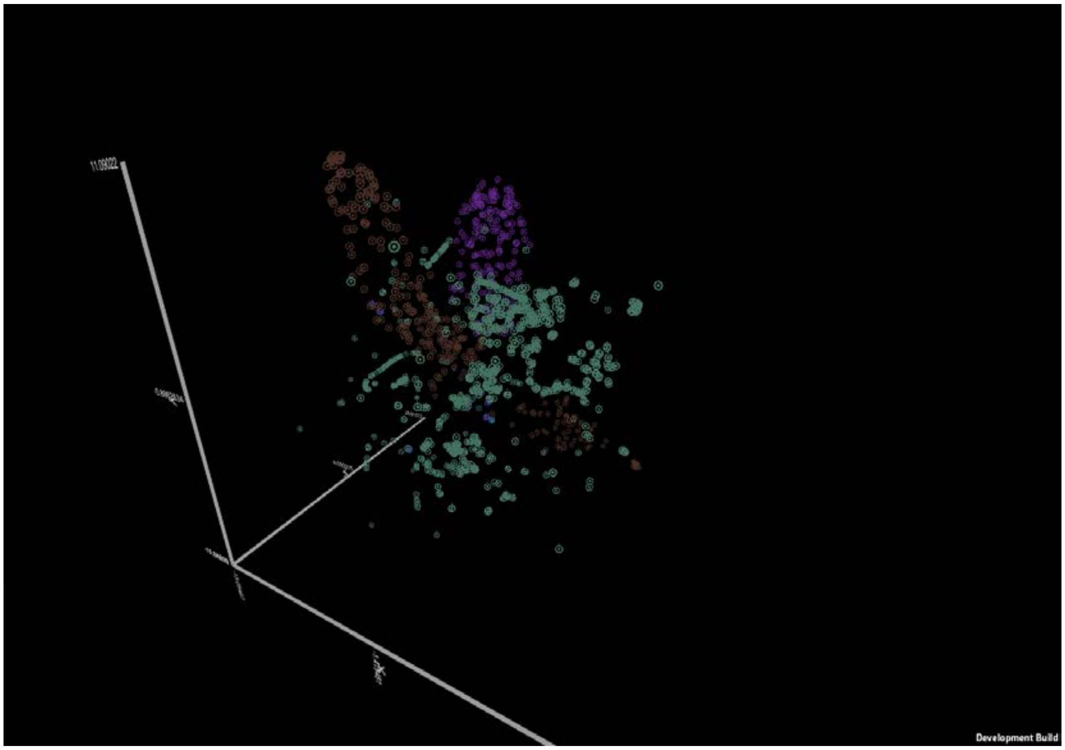


Fig. 10 Set of data points mapped to xyz coordinates by a t-distributed stochastic neighbor embedding conducted prior to data import into Vids

Vids offers some key advantages over traditional visualization methods in this task. Compared to a static rendered image produced by visualization reporting tools, Vids created visualizations that could be rotated and examined at varying levels of scale in real time. In addition, data in Vids became interactive—the plotted data can be inspected and manipulated exactly as the user desires. In addition, dynamic features such as the ability to change attribute to color mapping and the ability to change layouts are useful for reinterpreting the data without the need to restart the graphing program. It is possible, for example, to seamlessly change graph layouts from 3-D mapped points to chart layouts such as histograms and scatterplots based on user-selected features.

6. Conclusion and Future Development Plan

The goal of the development program is to provide a modern, highly flexible and functional research and development platform for 3-D visualizations relevant to network security and awareness. Vids' first intent is for research purposes. Vids allows a variety of visualizations to be constructed on a common platform, and these visualizations can be tested simultaneously with prospective users. Results from user evaluations of Vids can then be used as feedback for new development or refinement of network security visualizations. Once this research and evaluation phase is complete, Vids may move into active use as a situational awareness tool for cyber analysts that could be displayed on workstations, large screens, and/or set up as a training tool to interact and understand data in an easier-to-visualize way. Future development of post-alpha Vids includes the following new and expanded features aimed at reaching the goal of wider flexibility and functionality:

- Expanded options for input data sources. Compatibility with a number of different data sources is desired beyond the current local CSV reading capability. JSON reading is a likely next step, along with remote operations such as reading a remote file, accessing a database, or using a representational state transfer) application-program-interface-based service to retrieve updated data.

- Expanded user ability for the user to interact with data sources through Vids, and perform transparent configuration of data sources from within Vids. Data sources will be reread and the graph will be dynamically reconfigured to accommodate changes since the last reading and rendering.

- Novel layouts based on new algorithms or user feedback as to the most useful data views.

- New styles for nodes and edges to allow maximum options for data visualization.

- Expanded interaction options, including node/edge-to-graph translation. This feature would allow data contained within an individual node or edge to be shown and interacted with graphically rather than through text. For example, a timeline graph could be constructed showing activity over time for a given pair of connection partners. This new graph would be shown above or on the existing edge linking 2 nodes within a node-edge flow graph.

- VR/AR compatibility, or at minimum, transferability of component designs and knowledge transfer. Technically compatible now, but only useful for static display due to dependence on keyboard and mouse interaction.

The Vids project aims to demonstrate a new direction in 3-D interactive visualization for the Army. Faced with ever-increasing data volumes, new solutions are needed to maintain network situational understanding. Visualizations are one way to enable the Warfighter or network defender to process, and most importantly, understand, a larger volume of data. By using a modern game development platform, Vids allows streamlined development, strong portability across operating systems and platforms, and a variety of 3-D, VR, and AR display options. In summary, Vids is intended as a first step to bridge the gap between network and security visualizations as they currently exist and the envisioned future where visualizations act as a ubiquitous and crucial aid to operations in cyberspace.

7. References

Alberts D, Kott A, Rivera B, Chan K, Scott L, Hobbs R, Leung A, Dron W, Chadha R. Network science experimentation vision. Adelphi (MD): Army Research Laboratory (US); 2015 Sep. Report No.: ARL-TR-7451.

Ferebee D, Dasgupta D. Security visualization survey. In: Proceedings of the 12th Colloquium for Information Systems Security Education; 2008 June 2–4; Dallas, TX. CISSE: University of Texas, Dallas; c2008. p. 119–126.

Hu Y. Efficient, high-quality force-directed graph drawing. Mathematica Journal. 2005;10(1):37–71.

Shiravi H, Shiravi A, Ghorbani AA. (2012). A survey of visualization systems for network security. IEEE T Vis Comput Gr. 2012;18(8):1313–1329.

Trossbach LC Jr, Pino RE, inventors; ICF International Inc, assignee. Method and apparatus for visualizing network security alerts. United States patent US 9,142,102. 2015 Sep 22.

Wade N. The battle staff smartbook: doctrinal guide to military decision making and tactical operations. 2nd ed. Lakeland (FL): The Lightning Press; 2005.

Zage DM, Zage WM. Intrusion detection system visualization of network alerts. Muncie (IN): Ball State University; 2010. doi.org/10.21236/ada532723.

List of Symbols, Abbreviations, and Acronyms

2-D	2-dimensional
3-D	3-dimensional
AR	augmented reality
AREP	Applied Research and Experimentation Partner
ARL	US Army Research Laboratory
CSV	comma separated value
CyberVAN	cyber virtual ad hoc network
IDS	Intrusion Detection System
IP	internet protocol
JSON	JavaScript Object Notation
VIDS	Visual Intrusion Detection System
VR	virtual reality
VRDAE	Virtual Reality Data Analysis Environment

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIR ARL
(PDF) IMAL HRA
RECORDS MGMT
RDRL DCL
TECH LIB

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 ARL
(PDF) RDRL CIN D
G SHEARER